Мониторинг большого количества движущихся по карте объектов

Егор Маслов





Вам будет полезен этот доклад

- если вы работаете с геоданными в реальном времени
- работаете с GPS-телеметрией
- не контролируете клиентские терминалы и их качество
- храните историю перемещений





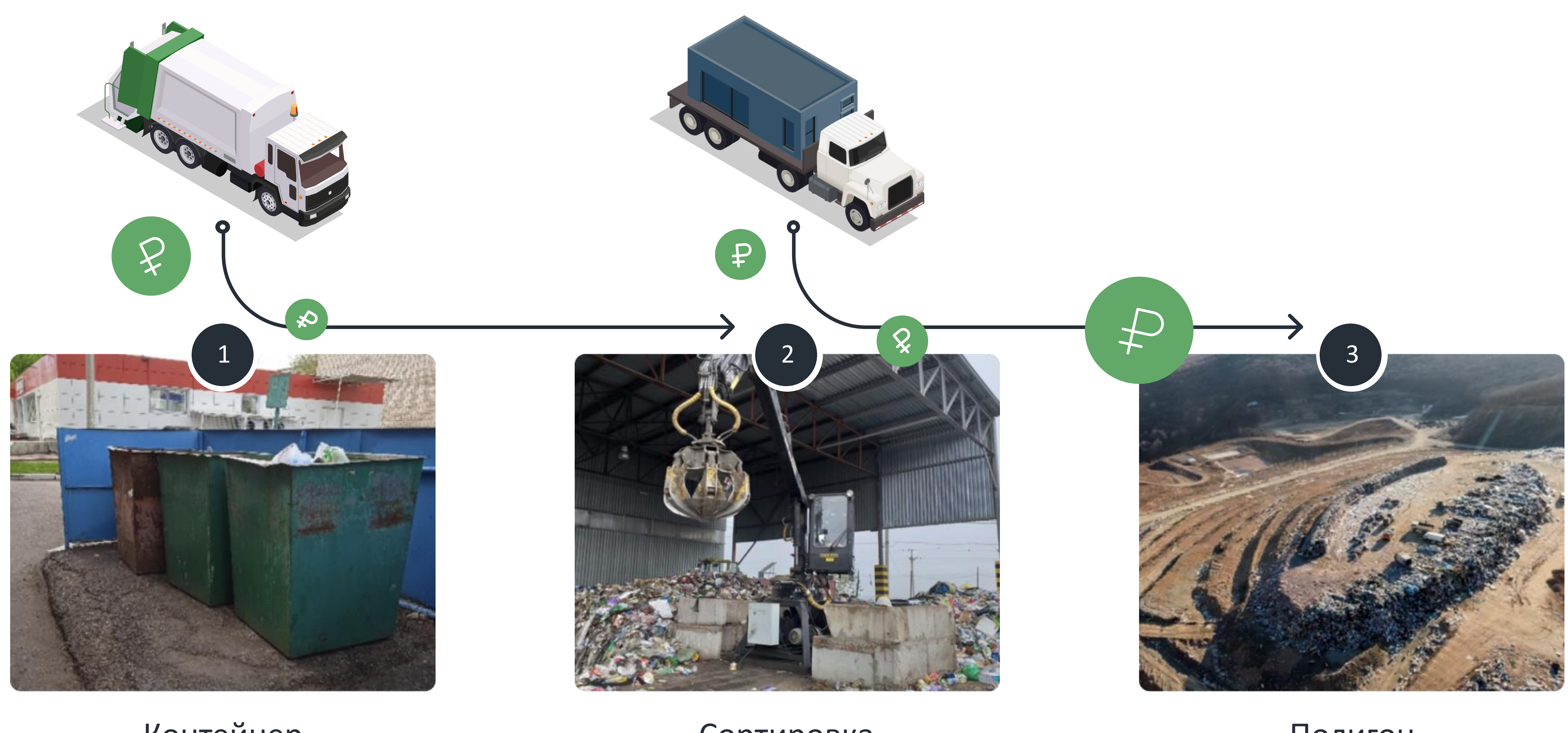

HL
HighLoad++
Весна 2021

1 Контейнер

2 Сортировка




1
2
3
Контейнер
Сортировка
Полигон


1
2
3
Контейнер
Сортировка
Полигон

При чем тут геопространственный мониторинг?

Аналитика и мониторинг позволяют:

- Выявлять неоптимальные маршруты и расположения объектов инфраструктуры
- Находить нелегальные свалки
- Следить, чтобы не переполнялись легальные
- Выявлять непокрытые маршрутами районы и населенные пункты
- Разобраться в причинах невывоза мусора
- Выявлять недостаточность контейнерных площадок



Очистка контейнеров



Потоки отходов



Компания «Большая тройка»

БОЛЬШАЯ
ТРОЙКА

5 лет разрабатываем информационные сервисы в области экологии

Случайный факт

Разработанные электронные модели оптимизации территориальных схем дали **6 000 000 000 ₽** экономии в год для операторов в области обращения с отходами.

Как результат

Снижение тарифов для населения в 24 регионах РФ



Как это работает

Камера

ГЛОНАСС,
GPS

LTE модуль

Нормальный интернет

RFID

Мобильное
приложение

40 000

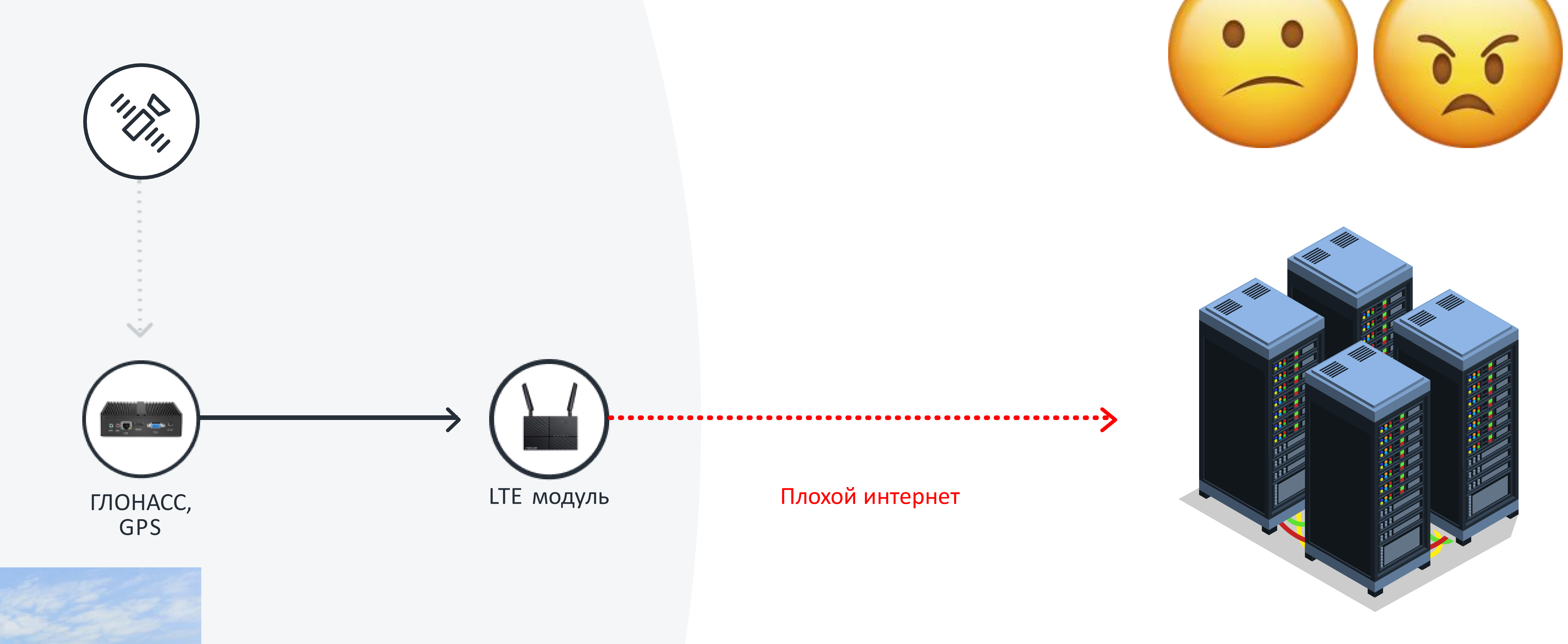
машин

8 000 000

объектов



Как это работает в некоторых регионах







Как мы это обрабатываем

40 000 машин

Зоопарк протоколов

Очереди

Система принятия решений, WebSocket

ClickHouse

- авторизация
- валидация данных

- обработка данных
- проверка прав
- фильтрация



Что будем обсуждать

- Проблемы входящего трафика
- Проброс информации о передвижении до браузера клиента
- Кластеризацию объектов, меняющихся во времени
- Создание карты покрытия
- И при этом попробуем поменьше программировать =)



- Проблемы входящего трафика
- Проброс информации о передвижении до браузера клиента
- Кластеризацию объектов, меняющихся во времени
- Создание карты покрытия
- И при этом попробуем поменьше программировать =)



Проблемы входящего GPS-трафика

- Дублирующиеся ID транспорта
- Устаревшие трансляции
- Мусорные данные (непрерывный ddos)
- Потоки архивных данных



Выход

- выделяем больше портов
- выделяем еще больше портов!



Масштабирование на входе

30001 → Haproxy / Traefik
30002 → Haproxy / Traefik
...
40000 → Haproxy / Traefik

Haproxy / Traefik → Тестовый режим → Приложение
Haproxy / Traefik → Боевой режим → Приложение
Haproxy / Traefik → Dev среды

```
elm.name
   {% endfor %}
   {%- if meta.balance %}balance roundrobin{% endif %}
{% endfor %}

{% for backend, binds in ports.items() %}
frontend proto_{{backend}}
   bind {{ ','.join(binds) }}
   mode tcp
   option tcplog
   default_backend {{ backend }}
{% endfor %}
```



- Проблемы входящего трафика
- Проброс информации о передвижении до браузера клиента
- Кластеризацию объектов, меняющихся во времени
- Создание карты покрытия
- И при этом попробуем поменьше программировать =)



Снижение нагрузки на бэк

40 000 машин → Зоопарк протоколов → Очереди → Система принятия решений, WebSocket →

Зоопарк протоколов:
- авторизация
- валидация данных

Система принятия решений, WebSocket:
- обработка данных
- проверка прав
- фильтрация



Geohash





Geohash



Geohash

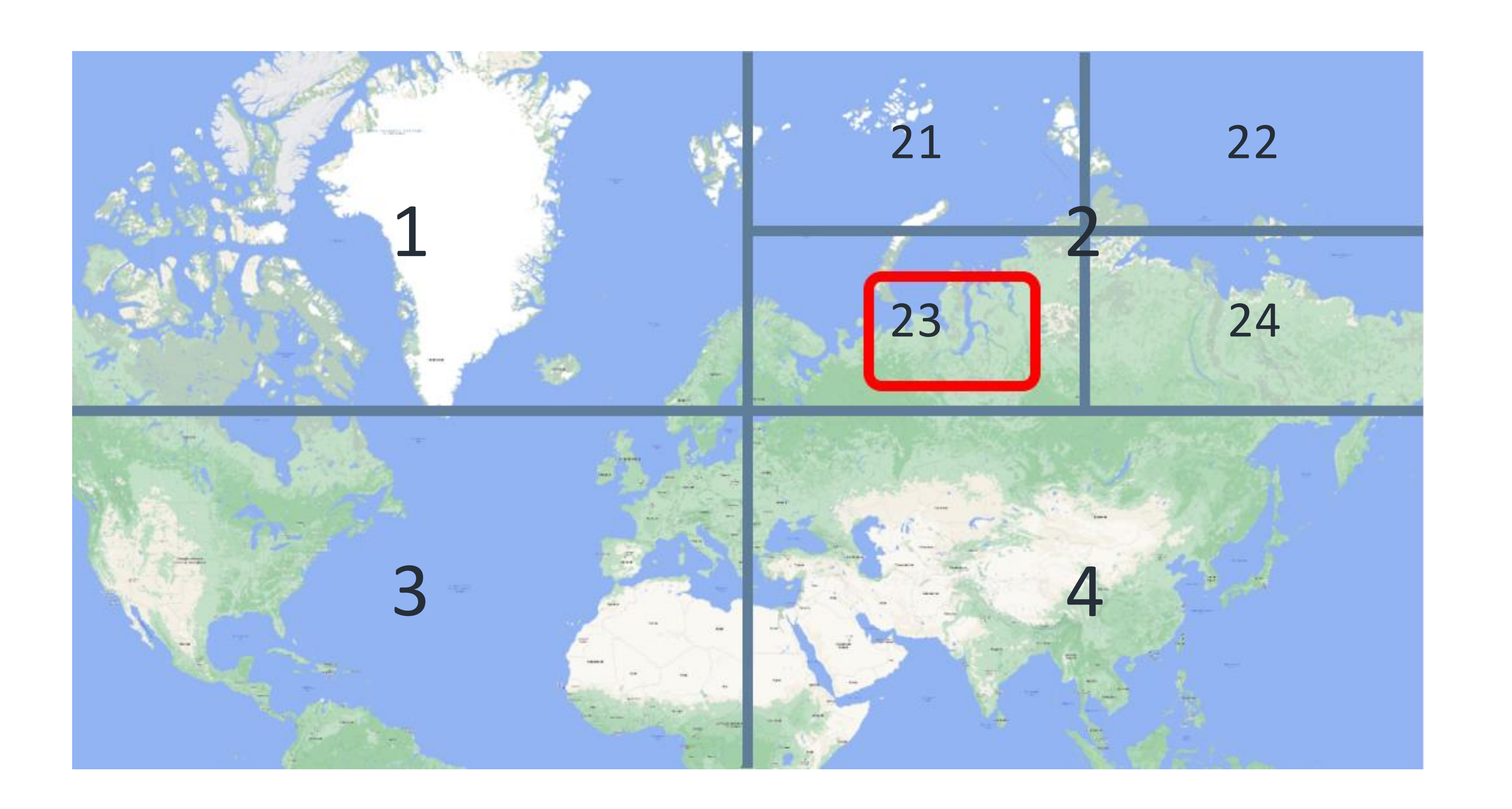
- Различные пропорции сегментов на проекции Меркатора

- Слишком большой шаг масштаба (1/24)



QuadHash

- Одинаковые пропорции и площади сегментов
- Шаг примерно равен шагу масштаба в leaflet





Используем геохэш в маршрутизации трафика

- Вычисляем geohash входящей точки (344562...23)
- Сообщение отправляем в обменник Rabbit с ключом 34.45.62....23
- Клиент сообщает бэку, что смотрит конкретную зону (напр 3445)
- Сервер подписывается на очередь с маской 34.45.*.*.*.*



Получилось!

40 000 машин → Зоопарк протоколов → Очереди → Система принятия решений, WebSocket →

- авторизация
- валидация данных

- обработка данных
- проверка прав
- фильтрация



Проблема граничных переходов

21
22
1
2
23
24
3
4



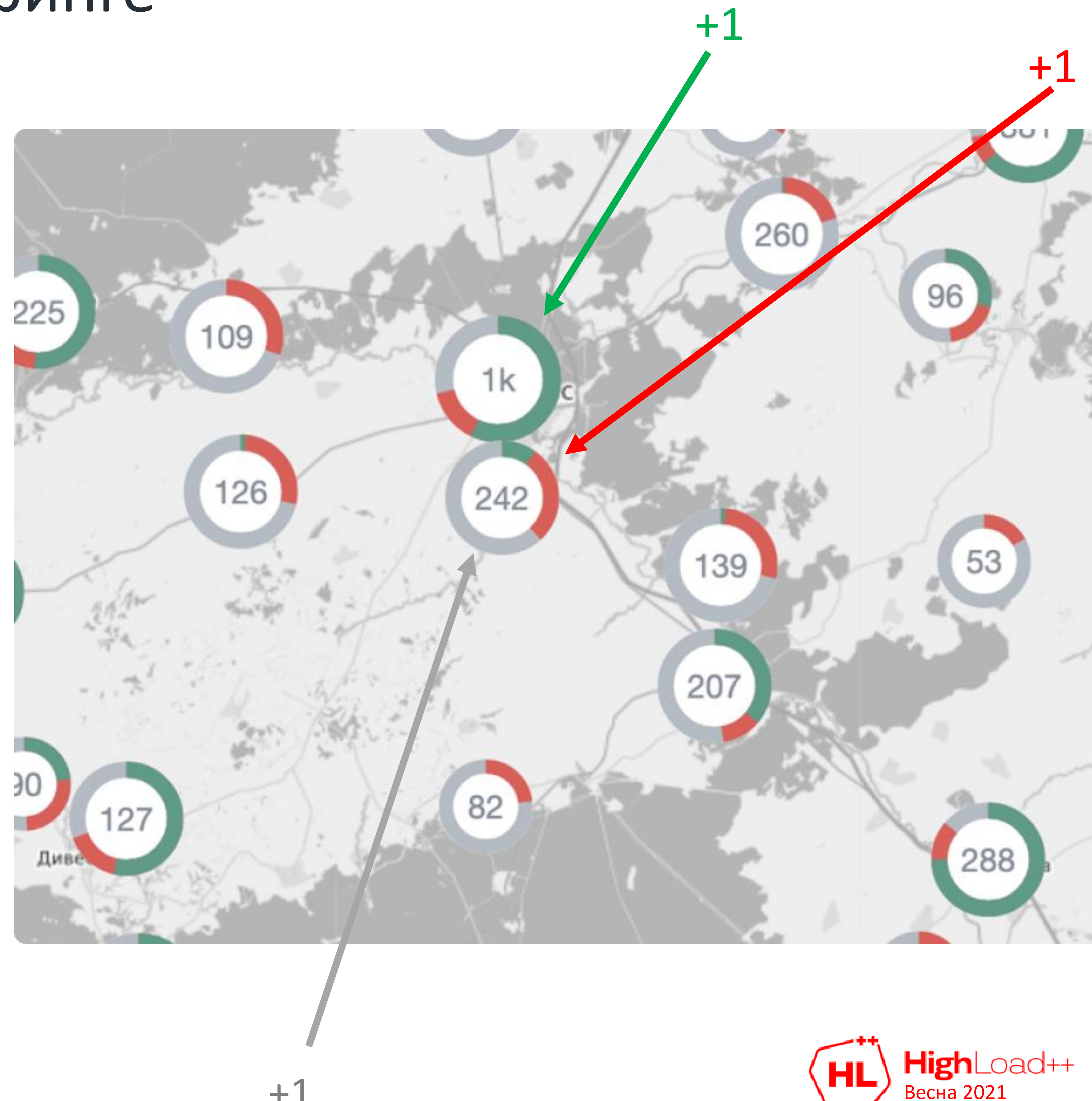
- Проблемы входящего трафика
- Проброс информации о передвижении до браузера клиента
- Кластеризацию объектов, меняющихся во времени
- Создание карты покрытия
- И при этом попробуем поменьше программировать =)



Кластеризация объектов в мониторинге

Данные постоянно меняются

- Кол-во точек
- Расположение точек
- Статистика событий по ним

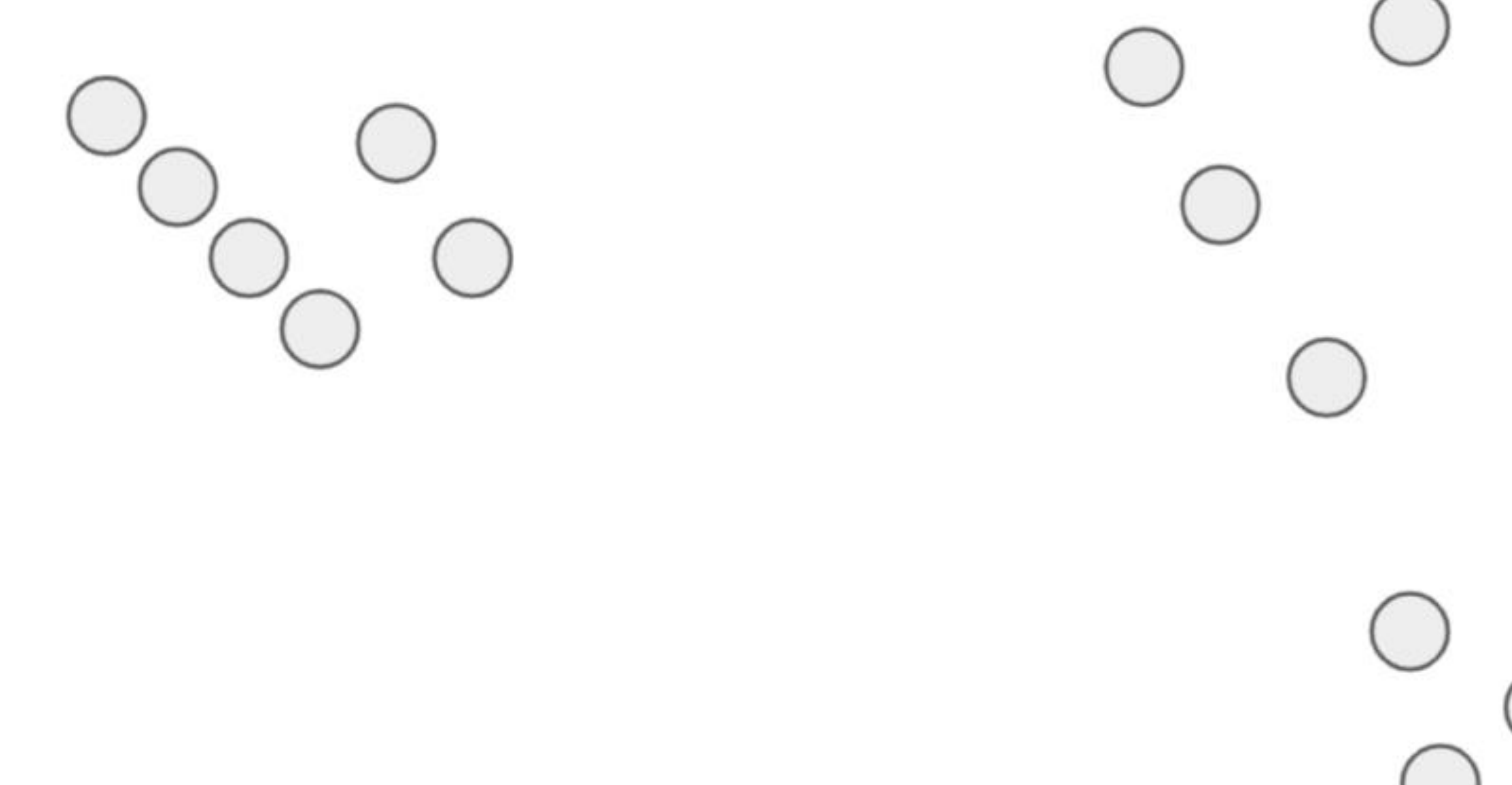


Алгоритмы кластеризации





Алгоритмы кластеризации





Алгоритмы кластеризации



Алгоритмы кластеризации



Алгоритмы кластеризации



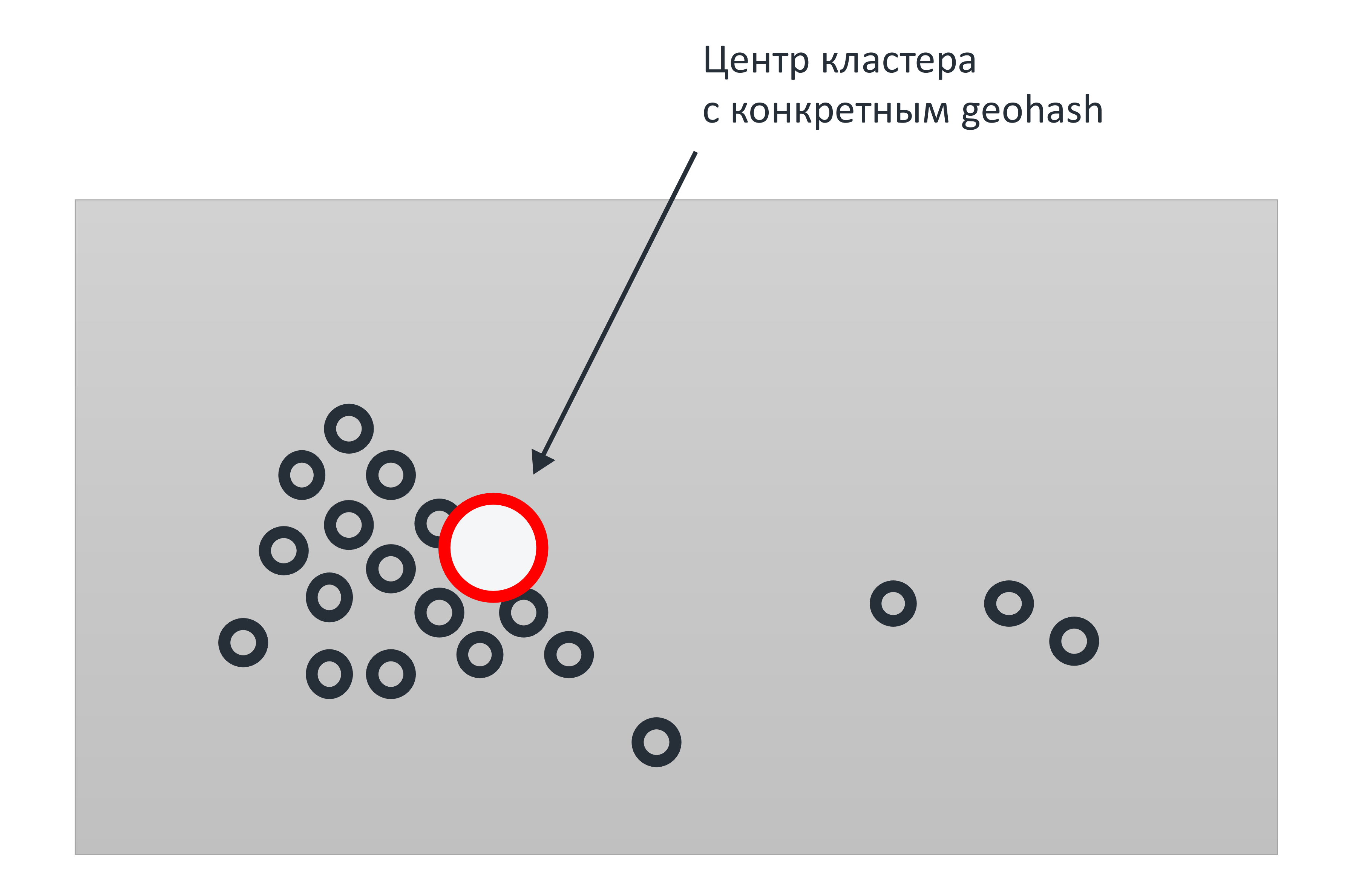
Вывод: кластеризация динамических объектов - ресурсозатратна

~~Кластеризация~~ Классификация



Геохэш-кластер

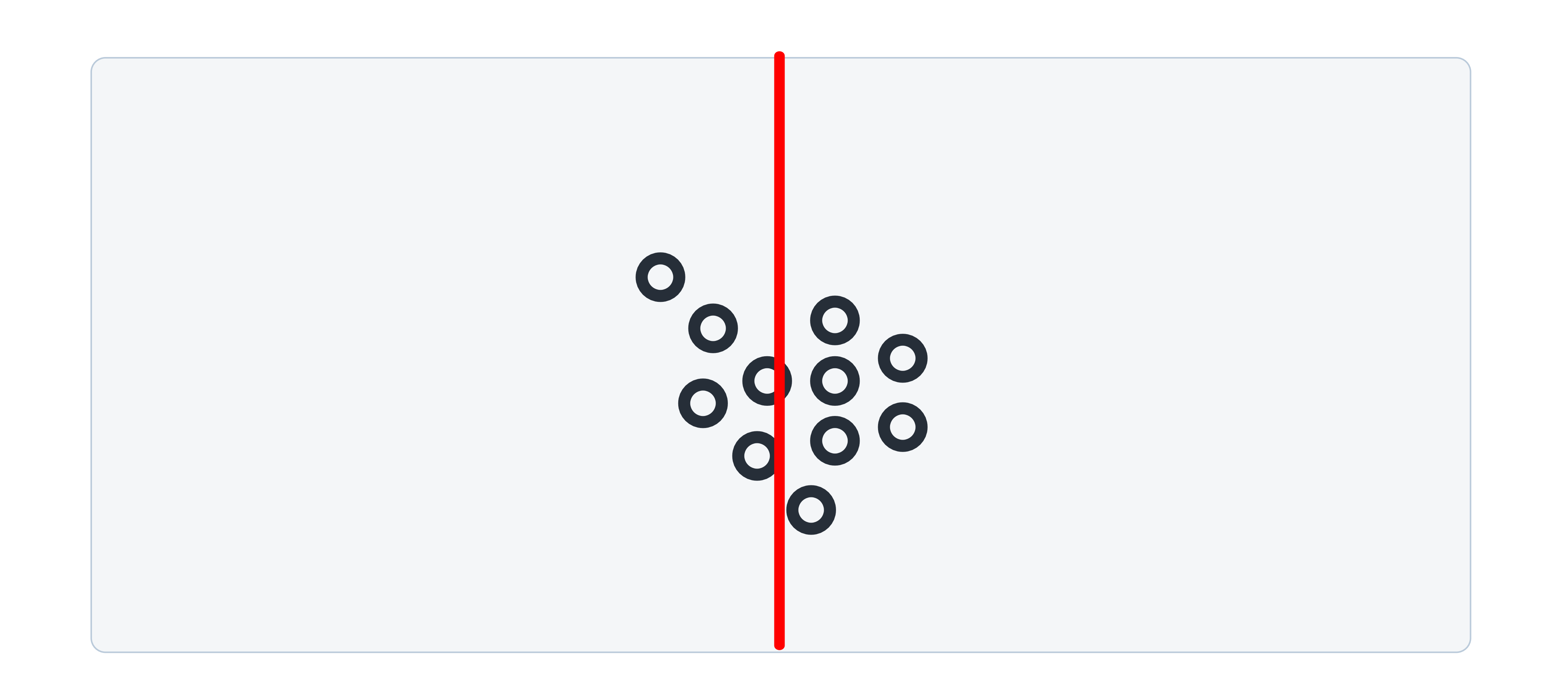
- находим все точки в квадранте
- выявляем центроид
- создаем кластер
- присваиваем ему хэш
- для каждого нового события находим кластер по простому индексу

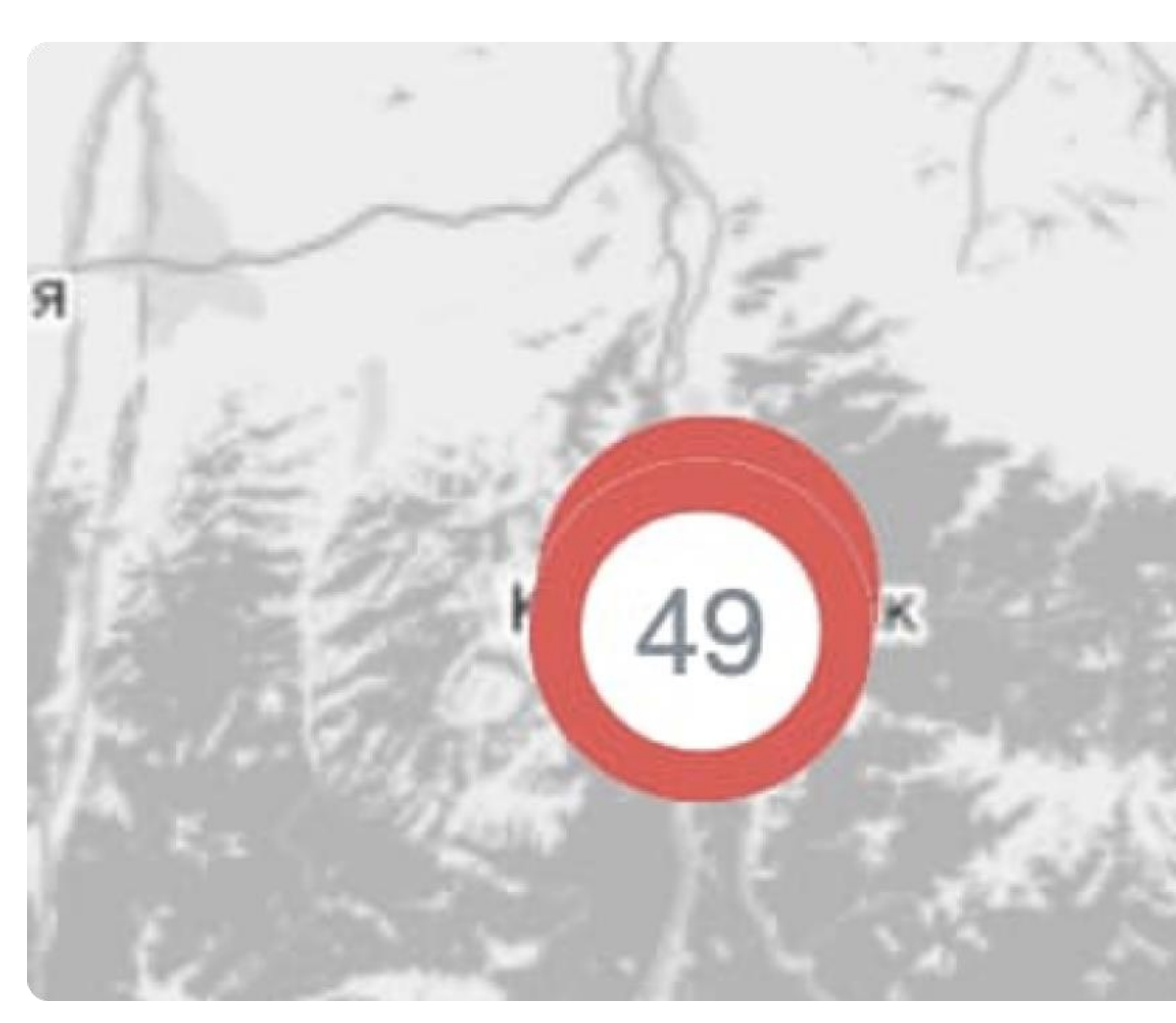




Геохэш-кластер > Проблемы

Срединное деление

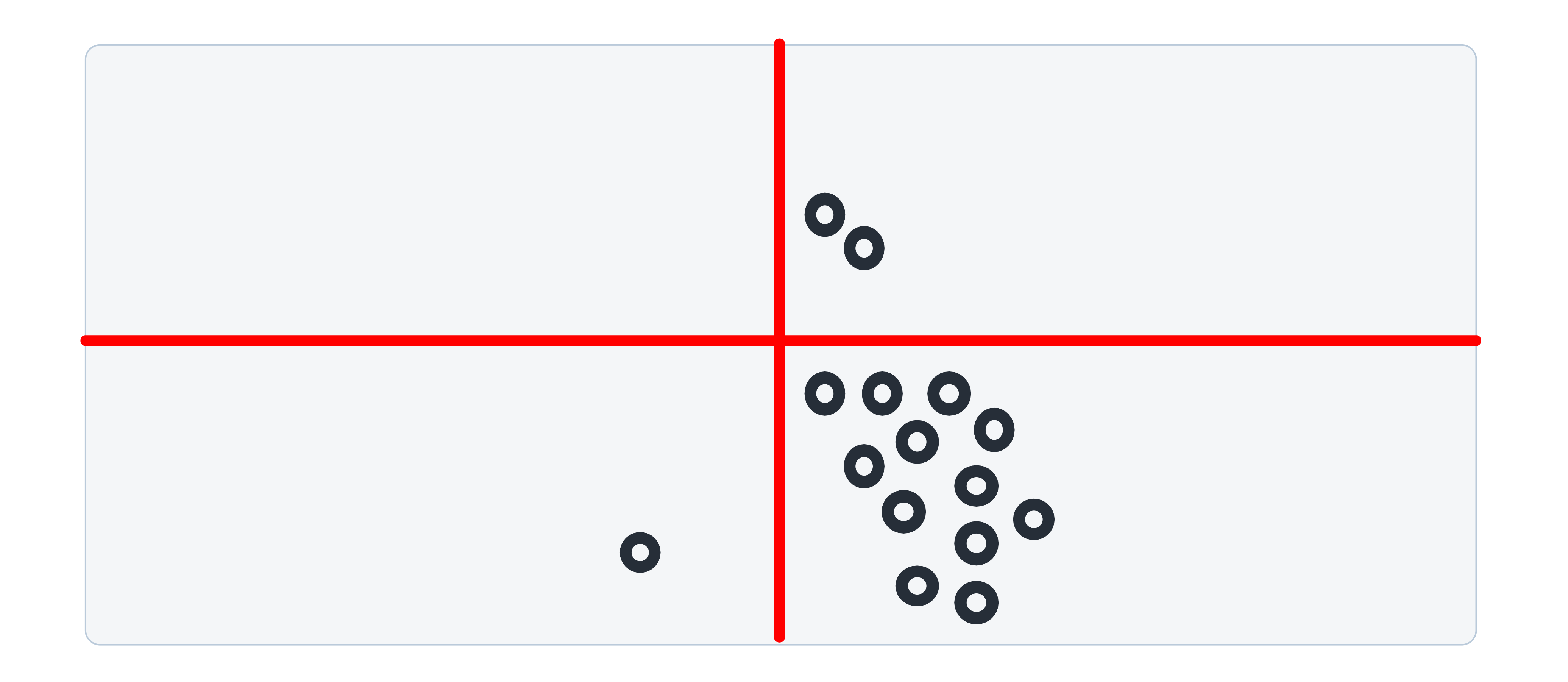


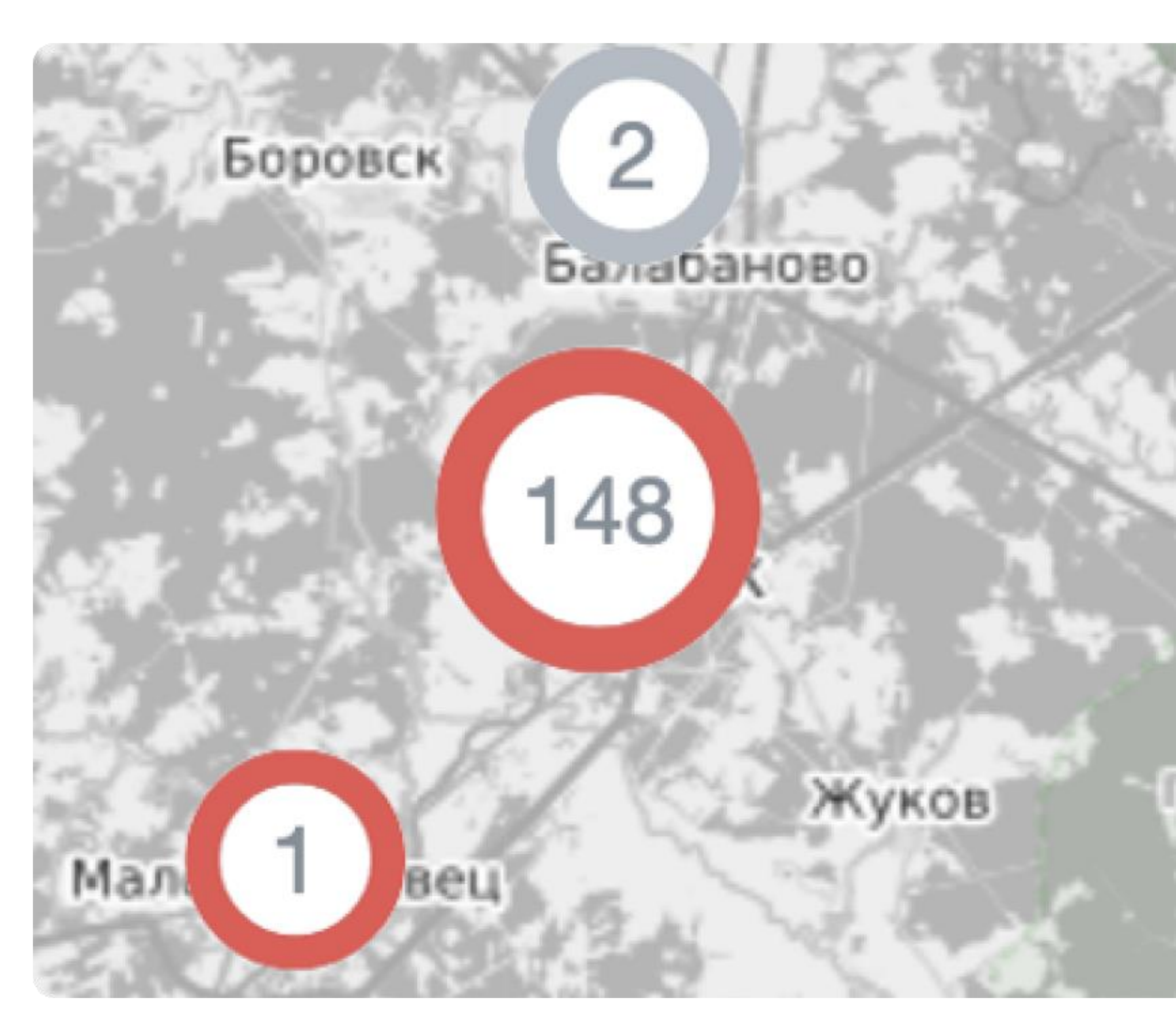




Геохэш-кластер > Проблемы

Проблема небольших остатков







Геохэш-кластер > Решение проблем

Легко решаются на фронте

в зависимости от того, на что делаем акцент







- Проблемы входящего трафика
- Проброс информации о передвижении до браузера клиента
- Кластеризацию объектов, меняющихся во времени
- Создание карты покрытия
- И при этом попробуем поменьше программировать =)



Карта покрытия в реальном времени

Tracker 1

Tracker 2

Tracker 3



3 вопроса к карте покрытия

1. Есть ли покрытие в данной зоне
2. Где конкретно кончается маршрут
3. Есть ли маршрут из зоны в зону





Почему не использовать граф дорог?

- Стандартный расчет маршрута искажает реальность

- Не везде где проедет мусоровоз можно проложить маршрут. У нас есть, например, катера ☺



Может просто отобразить маршруты?

25 s / 24 Mb



Возьмем 2 сходящихся маршрута



Которые хранятся в ClickHouse

	Lat	Lon	Timestamp	TrackerId	Qhash
1	66.53007998050425	66.67216595883299	2021-05-10 23:59:59	474625	10301113033010030120
2	55.33243942198633	37.54408260284552	2021-05-10 20:02:31	43886549	10231012121122301013
3	55.91033554075061	37.41963573205742	2021-05-10 20:02:31	39141047	10231010122331222020
4	55.91123198529501	37.423530551005044	2021-05-10 20:02:31	34883194	10231010122331221213
5	55.62837598743578	37.019855495779744	2021-05-10 20:02:31	33820877	10231010232100310302
6	55.91226958108886	37.42153167664575	2021-05-10 20:02:31	33389063	10231010122331202332
7	55.6281966985269	37.02035903861289	2021-05-10 20:02:31	31137720	10231010232100310321
8	55.912971481660605	37.83488460300371	2021-05-10 20:02:31	265008758	10231010133221313032
9	55.96500394771923	37.76232146140242	2021-05-10 20:02:30	43842005	10231010132112322310
10	55.706321714843234	36.22223660727549	2021-05-10 20:02:30	37568167	10231001330002122222



Вариант 1 – простая группировка

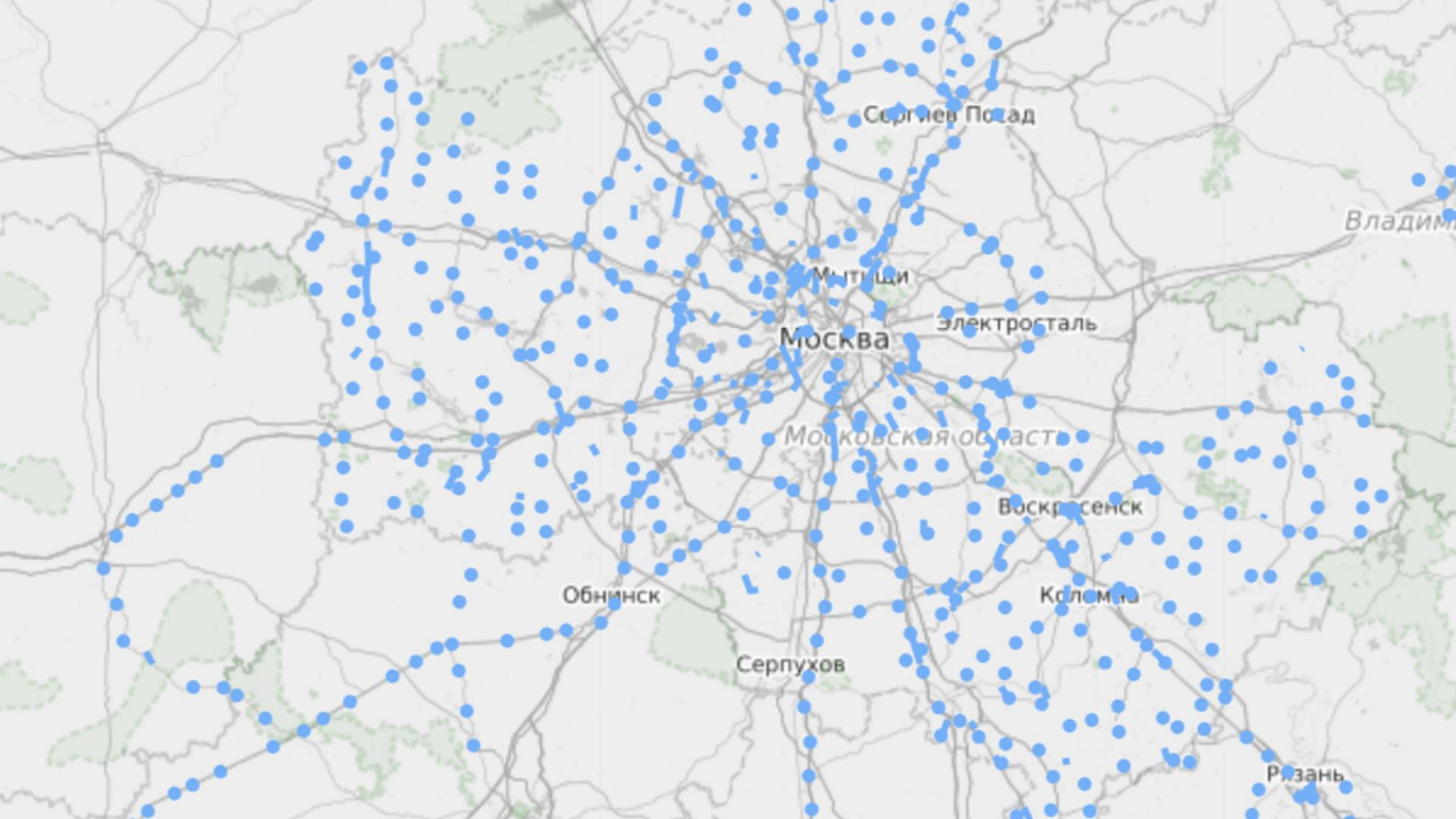


Вариант 1 – простая группировка

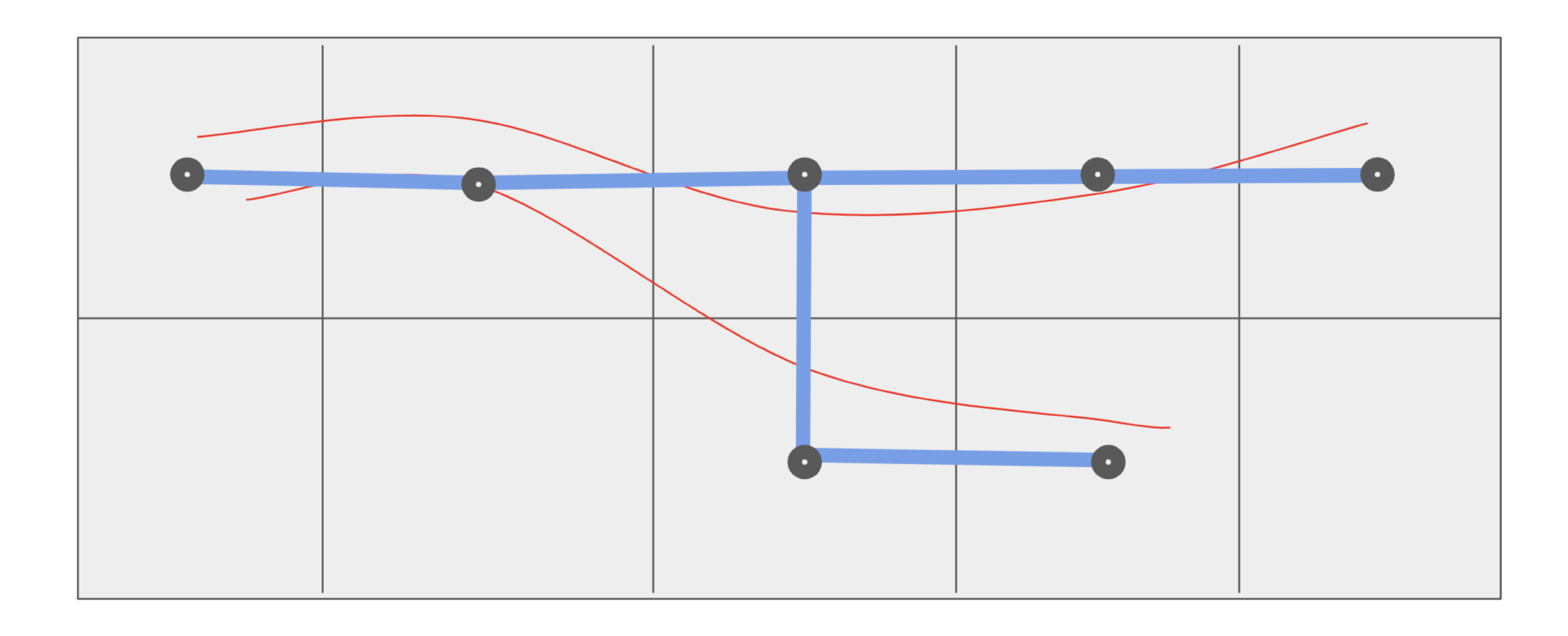

```
select
        avg(Lat), avg(Lon)
from routes
group by
        substr(
                  geohash,
                  1,
                  <zoomlevel> + 4
)
```

0.1 s / 84 Kb

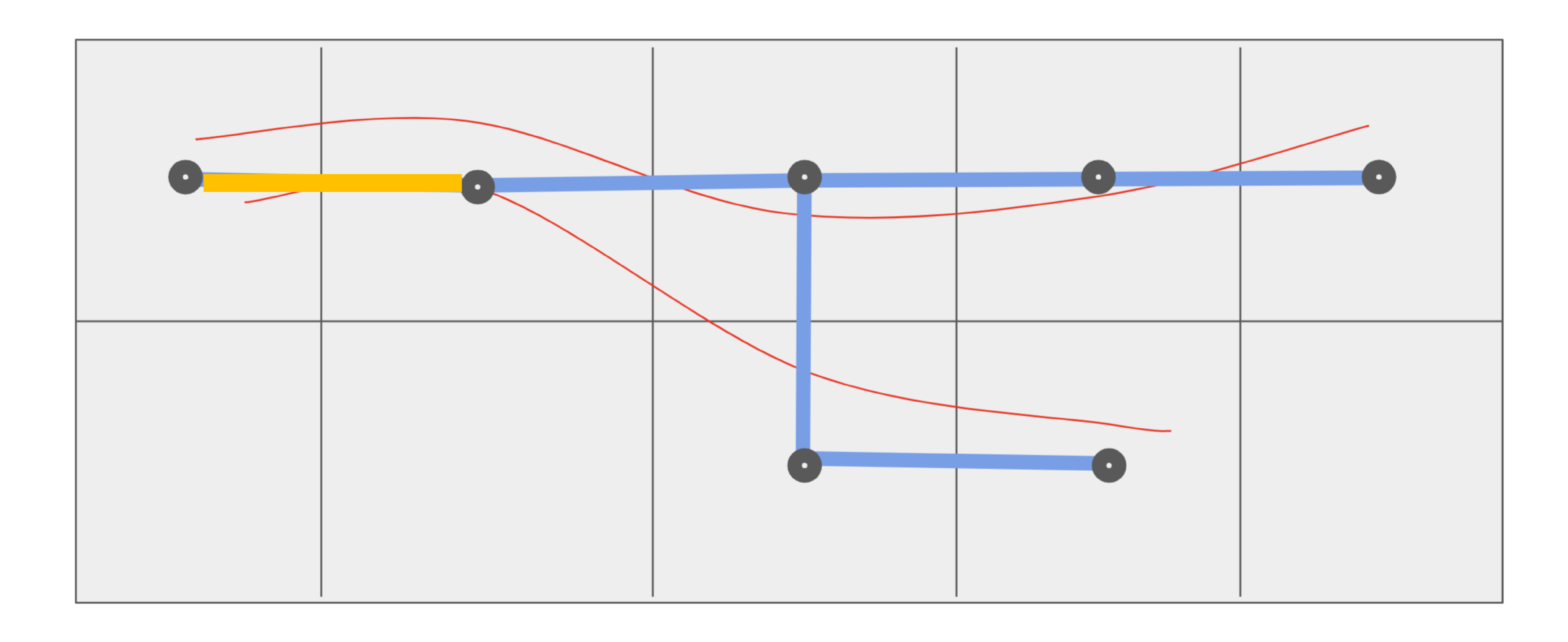



Сергиев Посад
Владими
Мытищи
Москва
Электросталь
Московская область
Воскресенск
Обнинск
Коломна
Серпухов
Рязань

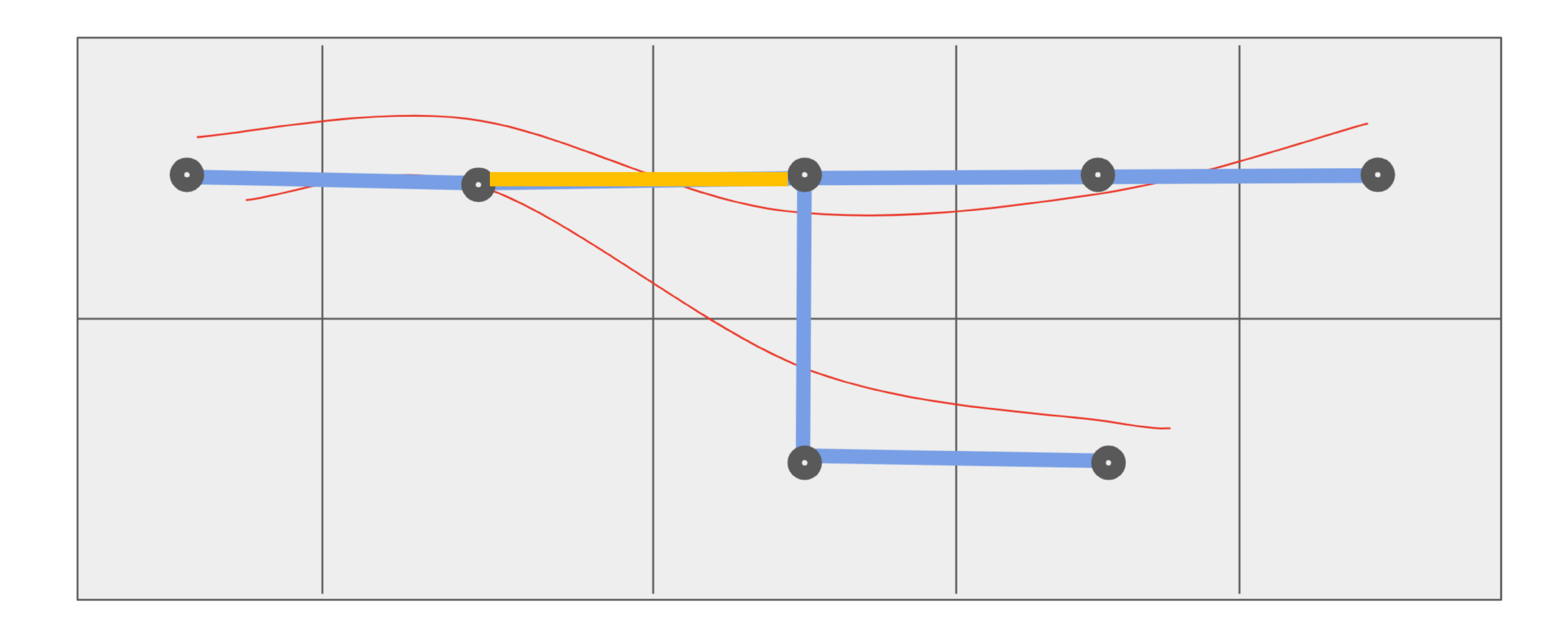
Вариант 2 – достаем из БД ребра графа



Вариант 2 – достаем из БД ребра графа



Вариант 2 – достаем из БД ребра графа



Вариант 2 – достаем из БД ребра графа


```
select
    hash,
    if (prev_tracker != tracker, null, prev_hash)
    from (
        select
            d.TrackerId as tracker,
            neighbor(d.TrackerId, 1) as prev_tracker,
            d.qhash as hash,
            neighbor(hash, 1) as prev_hash
        from (
            select TrackerId, Lat, Lon, substr(Qhash, 1, %(zoom)s) as qhash
            from routes
            order by TrackerId, Timestamp
        ) d)
    group by
        hash, prev_hash
```

Запросим упорядоченные линии маршрутов

(точки и их хэши)



Вариант 2 – достаем из БД ребра графа


```
select
    hash,
    if (prev_tracker != tracker, null, prev_hash)
    from (
        select
            d.TrackerId as tracker,
            neighbor(d.TrackerId, 1) as prev_tracker,
            d.qhash as hash,
            neighbor(hash, 1) as prev_hash
        from (
            select TrackerId, Lat, Lon, substr(Qhash, 1, %(zoom)s) as qhash
            from routes
            order by TrackerId, Timestamp
        ) d)
    group by
        hash, prev_hash
```

Создадим ребра между каждой парой точек, отметим те, у которых точки от разных автомобилей



Вариант 2 – достаем из БД ребра графа


```
select
    hash,
    if (prev_tracker != tracker, null, prev_hash)
    from (
        select
            d.TrackerId as tracker,
            neighbor(d.TrackerId, 1) as prev_tracker,
            d.qhash as hash,
            neighbor(hash, 1) as prev_hash
        from (
            select TrackerId, Lat, Lon, substr(Qhash, 1, %(zoom)s) as qhash
            from routes
            order by TrackerId, Timestamp
        ) d)
    group by
        hash, prev_hash
```

Удаляем повторяющиеся ребра, получаем пары a_hash -> b_hash, остается только найти их центроиды



Вариант 2 – достаем из БД ребра графа

0.5 s / 240 Kb





Зеленоград
Красногорск
Звенигород
Одинцово
Новый
Городок
Голицыно
Московский
Калининец
Селятино
Троицк
Можайск

Вариант 3 – используем средние координаты



Вариант 3 – используем средние координаты


```
select a_lat, a_lon, b_lat, b_lon from <prev_query> s

inner join (select

avg(`Lat`) as a_lat, avg(`Lon`) as a_lon,
       substr(Qhash, 1, %(zoom)s) as hash

from routes
            group by hash) d on d.hash = s.hash

outer join (select

avg(`Lat`) as b_lat, avg(`Lon`) as b_lon,
       substr(Qhash, 1, %(zoom)s) as hash

from routes
            group by hash) b on b.hash = s.prev_hash
```

Накладываем на предыдущий вариант средние точки маршрутов по всем транспортным средствам



Вариант 3 – используем средние координаты

1 s / 820 Kb




Дедовск
Новый
Городок

Вариант 3 – Проблемы

Незакрытые петли



Вариант 3 – Проблемы

Ложный ответ про маршрут «из зоны в зону»

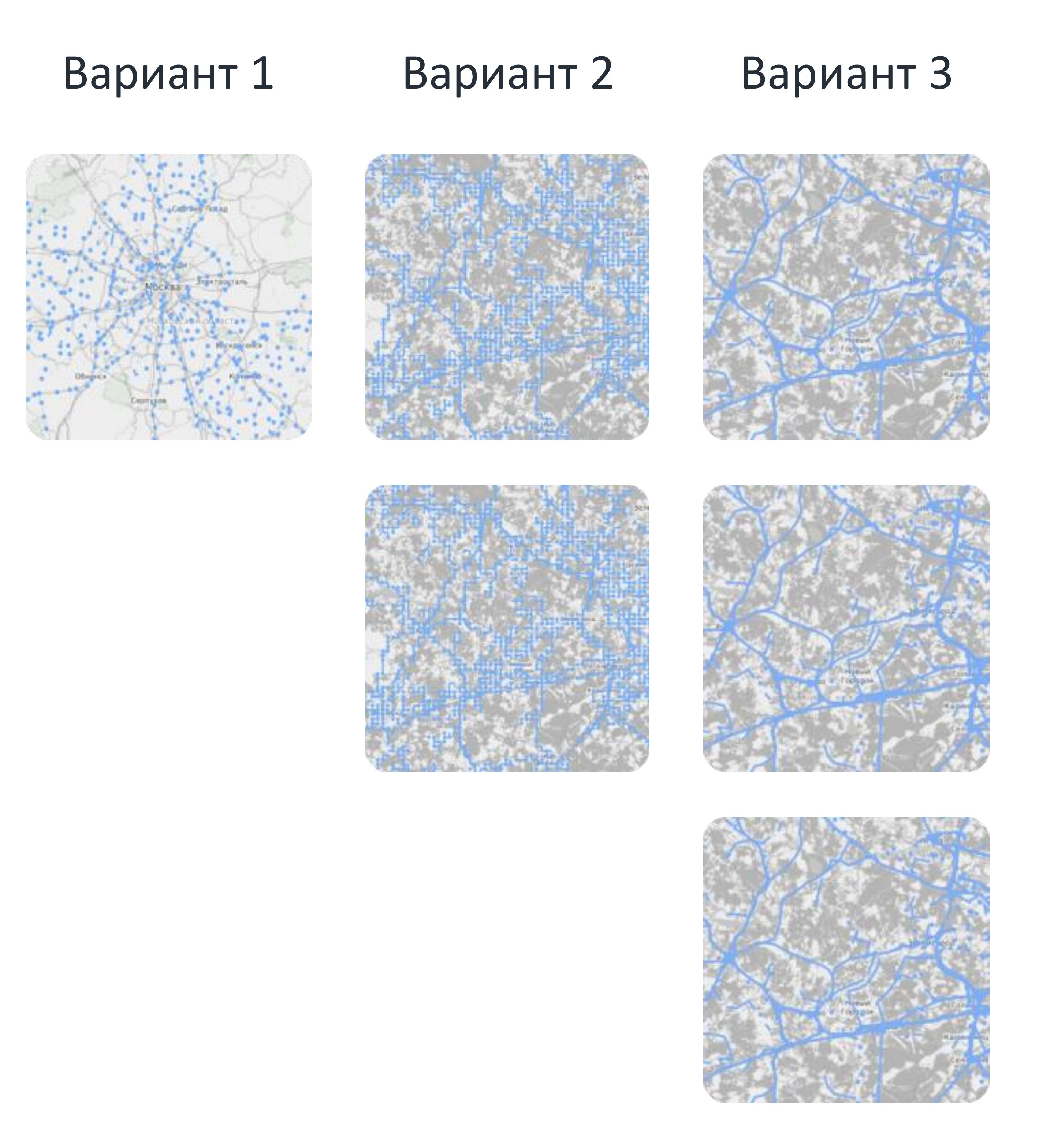


Карта покрытия – итоги

1 Есть ли покрытие в конкретной зоне

2 Где конкретно кончается маршрут

3 Есть ли маршрут из зоны в зону*





Как это выглядит



Как это выглядит

Жуков
Государственный природный заказник федерального значения Государственный комплекс «Таруса»
Оболенск
Серпухов
Протвино



Как это выглядит



Как это выглядит



Включаем кэширование

Тоже по геохэшам



Что и как мониторить в телеметрии

- Объемы трафика
- Кол-во пакетов
- Кол-во ошибок валидации протокола
- Запаздывание пакетов
- Очереди



В общем, как работать с геоданными в реальном времени:

1. Выделяйте как можно больше портов
2. Считайте geohash при записи в БД
3. Отдавайте маршрутизацию на уровень очередей
4. Не используйте стандартный геохеш и кластеризацию
5. Используйте clickhouse для хранения маршрутов





Спасибо за внимание!

Егор Маслов

Telegram: @gvamm3r

gvamm3r@gmail.com

